

Synchronization of the GPS Coordinates Between Mobile Device and Oracle Database System

Tomas Vana¹, Jiri Zechmeister¹, David Zak¹, and Jiri Lebduska¹

¹ University of Pardubice/Faculty of Electrical Engineering and Informatics, Pardubice, Czech Republic

Email: david.zak@upce.cz Email: {jiri.zechmeister, tomas.vana, jiri.lebduska}@student.upce.cz.

Abstract— The article describes an architecture and implementation of module for acquiring a synchronization of GPS data between mobile device and central database system. The process of data exchange is inspired by SAMD algorithm. The article sequentially presents a solution of individual system components. Special attention is paid to the exchange data format. The processing of the exchanged data is also described in detail. The resulting solution was deployed and tested in a real production environment.

Index Terms— GPS, SAMD algorithm, synchronization, Oracle Database, web application, Android platform, PL/SQL, XML, Google Maps

I. INTRODUCTION

There are many human activities which are carried out in a terrain form nowadays. Many employees do not perform their work in one central place. In this case, it is nice to have a tool to monitor and coordinate the movement of employees which are work in terrain. The significant part of the population currently owns the mobile device which offers enough performance and sufficient technical equipment (like a GPS module). So, the technical equipment is not the obstacle in principle.

The main problem is how to effectively a safely record, transmit and process these positional data from the mobile device to the central database system. Every interaction between mobile device and database must be realized with regard to costs of data transfers and with regard to limited battery capacity of mobile devices. Next key challenge is processing of the data itself on the database side. The main goal of this article is present the solution for employees tracking which is based on effective synchronization of GPS coordinates between mobile device and central database server.

The article has the following outline. The following section briefly presents the whole system. This will allow to better understanding module context within the whole system. The third chapter examined in detail architecture of the module for synchronization of GPS data. The conclusion is devoted to the presentation of positional data to users. The conclusion also includes brief evaluation of production deployment.

The rest of this paper is organized as follows. Section two presents some of the related work. Section three introduces the system context. Section four describes the details of the synchronization and section five describes the visual interface of whole system. Section six shows experimental results and section seven introduces the discussion. Section eight concludes this paper.

II. RELATED WORK

At first it's good to say that the development of whole system was driven by specific requirements and the whole system was built up from scratch. But the design of the resulting system is inspired by several existing

solutions. For example we were inspired by Alora and from the local system we were inspired by system called “Pečovateľská služba”.

The system was developed as cloud-like application. There are several articles about development of cloud application, about the economics of cloud [14, 15], about future [16] of cloud application and about cloud applications security [17].

Another challenge was to develop a mobile subsystem. There are several ways to implement this subsystem. First way is to implement the mobile application as web application optimized for mobile devices [18]. Another way to implement a mobile subsystem is to create stand alone mobile application which is periodically synchronized with central database system [19, 21]. There is also exist combination of the web-based and stand-alone application [20].

III. SYSTEM CONTEXT

The employees tracking (with synchronization) feature is not stand-alone application. This feature was developed as a module of the modular application called TOPS. This chapter briefly describes this modular application and tries to set up the context for the usage of tracking module.

The main aim of TOPS application is to provide complex information system for organizations providing terrain social services [1][2]. The application tries to cover all of basic needs of these organizations such as:

- client evidence,
- care orders in the form of vouchers or contracts,
- staff evidence,
- scheduling clients' visits,
- data acquisition about provided services directly at clients' home,
- care reporting,
- invoicing of provided services to health insurance companies or to clients.

The whole system can be divided into two parts from the users view. The first part of the system is represented by mobile subsystem. Each terrain worker has its own mobile device (Samsung smart-phone with Android operating system) with pre-installed TOPS mobile application (Fig. 1). The TOPS mobile application primarily provides daily overview of the planned activities for the workers. Mobile application is also used for reporting of the worker's daily activities

The worker's position tracking runs in background of the application. The recording is fully automated and absolutely transparent for the user. The collected positional data are periodically sent to the central database system.

The system can be divided into three parts (tiers) from developer's point of view (Fig. 2).

First tier is represented by mobile subsystem. The smart-phone with Android OS is core part of mobile subsystem. In our system we are using smart phones with Android system (version 2.1 and 2.3). These mobile devices communicate with central database system by cell network. GPRS and EDGE are typically used to transfer data from mobile device to the database system.

A. Basic system architecture

As a second tier of the system we can mark the central database system. This central database system has two main responsibilities. First, it is the central data storage for all application modules. Second, the whole business logic is implemented there in form of stored PL/SQL procedures. On the central database system there is installed Oracle Database 11g in Standard Edition.

The third tier is represented by modular web application. This web application is used for complex management of the whole system. The web application is designed primarily for coordinating staff and managers. The advantage of the web solution is the ability to use the application practically from anywhere and from any platform. This web application is implemented by standard set of web technologies such as PHP, JavaScript and HTML with CSS.

IV. SYNCHRONIZATION OF GPS COORDINATES

In the development phase we consider many ways of location and tracking of mobile devices. We consider the pure GPS module, triangulation, wireless LAN and etc. A brief summary of the technologies applicable to tracking of mobile devices is in [4]. Finally, we choose the GPS module way of collecting coordinates.

Despite some limitations such as signals are subject to interference from atmospheric conditions, buildings and trees, and the time to achieve a fix on enough satellites the GPS is still the best solution for mobile device tracking in our country.

The process of GPS coordinates synchronization with the central database system starts when user of mobile device lunches the appropriate TOPS application¹. The whole synchronization process (from position acquisition to processing of date by database) is shown in Fig. 3. Synchronization can be divided into five basic steps:



Figure1. Sample screens from the mobile application.

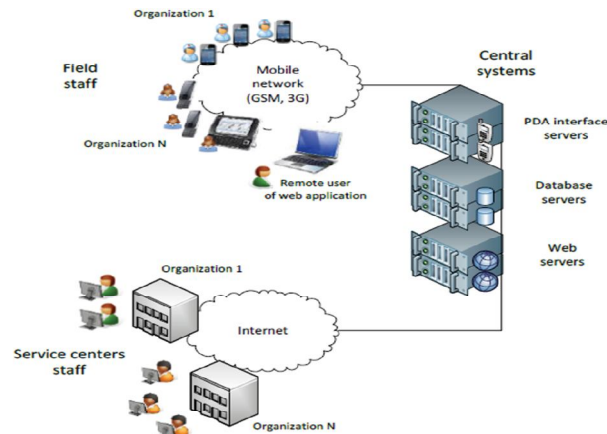


Figure2. Overall system architecture.

Mobile device records the position periodically (usually 10-60 seconds interval) using integrated GPS module.

2. Acquired GPS coordinates are not immediately sent to the central database, but coordinates are persistently stored in a local database (in our solution we use SQL Lite database) of the mobile device. This way solves several problems[3, 5].

It may happen that the mobile device will be temporarily out of GSM network or there is any other reason which makes data transfer impossible [6, 7]. Furthermore, the persistent storage in the mobile device eliminates the problems associated with unexpected technical problems (application crash, discharged battery, hardware failure ...).

In addition, initiate data transfer due to every newly acquired GPS coordinate would by both highly energy-consuming and at the same time transferring small data amounts is generally less effective (compared to batch transmission). That is the reason why all data transfers are realized in batches (batch is generated at periodic intervals from one minute to five minutes).

3. Any exchange of data between a mobile device and central database system uses XML structure. Every XML document generated by mobile device contains only the data which has been modified since the last synchronization. Mobile device stores the time (timestamp) of the last successful synchronization.

4. The generated XML document is sent through the mobile network to a PDA interface server.

5. After processing the request the PDA interface server sends the XML document to the central database system where the XML document is processed.

A. Structure of the XML document

The communication between mobile devices is based on exchange of two types of XML documents. These documents are called as "Request" and "Result" documents. The request document is created on the mobile device. Result document is generated on the database side as a response on the request document. The response document will be introduced first. Sample request document is presented in Fig. 4.

The content of this document can be divided into two parts. First part can be labeled as Authentication header (element USER_IDENTIFICATION). Second part of the document then contains the actual positional data (element LOCATIONS inside element TABLE).

¹This statement is not entirely accurate. Some of the organizations do not use employees position monitoring. There is no GPS synchronization in this case.

The simple authentication is used in case of GPS coordinates synchronization. So, the IMEI number is the only content of the USER_IDENTIFICATION element (system also support full authentication; more about full authentication in [8]). Simple authentication is sufficient for location synchronization because the data transfer is one way (from mobile device to the central database) only and no sensitive data are sent back. Data related to the location are transferred inside the LOCATION element. Individual location information is contained in element ROW which technically represents a row in database table in local and central database.

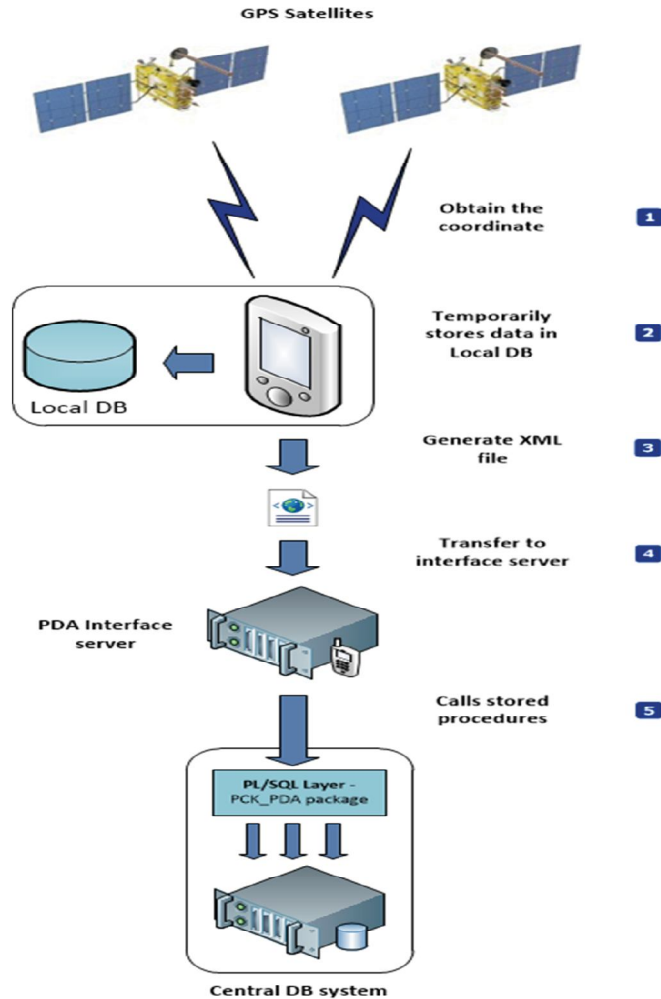


Figure3.GPS coordinates collecting and processing

Important attributes of the ROW element:

- loc_latitude – latitude.
- loc_longitude – longitude.
- loc_accuracy – inaccuracy of GPS module.
- loc_provider – provider of positional data (G - GPS, N – network, triangulation).
- loc_status_mesage – tells why the position was recorded (0 – time or distance limit exceeded, 1 – GPS provider change status).
- provider_status – information about provider status.
- loc_sample_timestamp – UTC time stamp
- event_message – additional information about position.

The sample request document shows which information about location is recorded. It is obvious that not only location information (longitude, latitude) is recorded and not all row elements contain valid GPS coordinates.

This additional information is very useful for troubleshooting and this information is not available for the regular users.

One of the disadvantages of XML documents is their size in compare to other technologies (like JSON). So, the XML document is compressed before sending.

The G-zip compression based on Lempel-Ziv and Huffman coding method was chosen as compression method. This method is very powerful in combination with XML and the method is implemented in most platforms.

B. XML document transfer into central database

The mobile device sends the document to the PDA interface server after its creation. Then the interface server decides to which database sends the request XML document (there are many database servers in the system – production, testing and development database).

The database is accessed through isolated database schemes due to security reasons. These schemas are very simple and contain no data at all and have controlled access (using business rules) to database tables. In case of unauthorized access attacker does not obtain access to sensitive data.

C. Scheme of central database system

The whole principle of interaction between the mobile device and the database system is based on the fact that the data within the database is accessed via isolated database scheme. For mobile device we called the scheme as TOPS_PDASERVICE. This scheme has no direct access to real data. This schema must access data through stored procedures in TOPS_SYSTEM scheme (Fig. 5).

```
<REQUEST>
  <USER_IDENTIFICATION>
    <MDE_IMEI>355299032267998</MDE_IMEI>
  </USER_IDENTIFICATION>
  <TABLES>
    <LOCATIONS>
      <ROW usr_id="441" work_status="1"
        loc_latitude="0,00000" loc_longitude="0,00000"
        loc_accuracy="0" loc_provider="G"
        loc_status_msg="1" loc_sample_timestamp="null"
        provider_status="AVAILABLE" loc_status="0"
        event_message="Provider's status has changed"
        created="2013-06-18 08:38:10" />
      <ROW usr_id="441" work_status="1"
        loc_latitude="50,03522" loc_longitude="15,794097"
        loc_accuracy="0" loc_provider="G" loc_status_msg="0"
        loc_sample_timestamp="2013-06-18 06:38:14"
        provider_status="Fix: 5/10 SNR: 270,160,210,120,130"
        loc_status="1" event_message="null"
        created="2013-06-18 08:38:16" />
    </LOCATIONS>
  </TABLES>
</REQUEST>
```

Figure4. XML for location synchronization (Request document).

So, the flow of request is following:

1. PDA Interface server is connected directly to the TOPS_PDASERVICE scheme.
2. This scheme has access to the package PCK_PDA in TOPS_SYSTEM scheme. TOPS_SYSTEM scheme has full access to data.
3. The synchronization calls stored procedures within the PCK_PDA package.
4. Procedures process the XML synchronization document.

There are two schemas which contain application data: TOPS_ADM a TOPS_SYSTEM², Within these two schemas are then implemented the tables form the diagram in Fig. 6. For basic position synchronization is very simple. There are only four tables by default:

- LOCATIONS – stores all information about each individual recorded position

²Data are stored in the database under two different schemes. TOPS_SYSTEM contains system data which are maintained by system administrators and TOPS_ADM contains data generated by the users of the system. This division is convenient for data backup.

- **USERS** – User’s details. Only users in this table can synchronize the position
- **MOBILE_DEVICES** – evidence of mobile devices
- **SYNC_XML_HISTROY** – synchronization log. Stores the raw XML synchronization documents for three week. Useful for troubleshooting.

Procedures inside PCK_PDA package expecting the XML document as an input. Once the document is received the document processing starts immediately. The procedures send back another XML document after processing the request XML document. This result XML document informs whether the processing of the request document was successful.

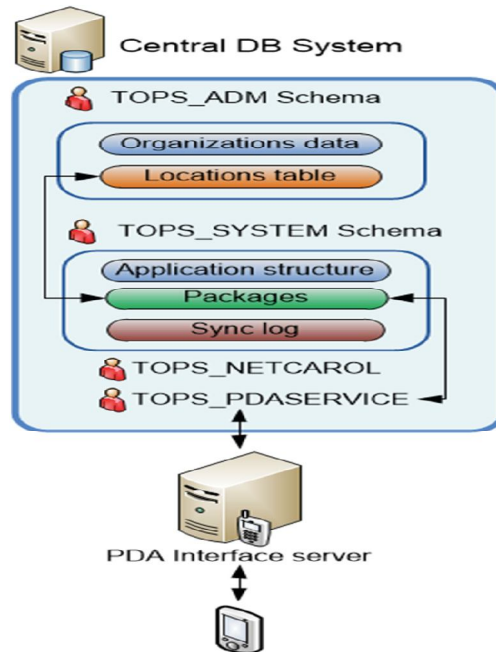


Figure5.Data workflow from mobile device to central DB

D. Processing of synchronization XML document

This XML result document is very important for mobile device. The mobile device set a new time stamp of last synchronization in case of successful processing of request document. Next synchronization will send only date that has changed after this timestamp. Example of result document you can see in Fig. 7.

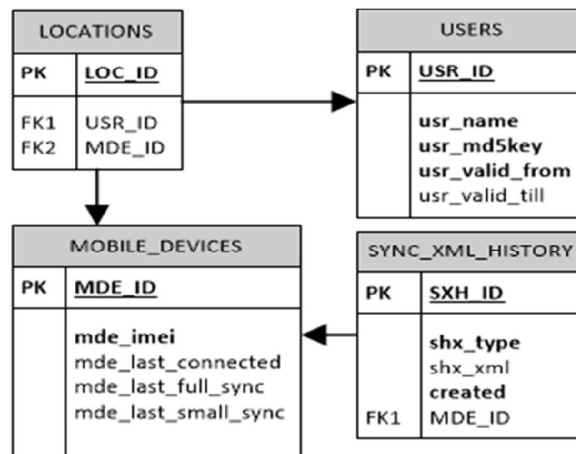


Figure6. Table scheme for GPS coordinates synchronization


```

<RESULT>
  <USER_IDENTIFICATION>
    <MDE_IMEI>355299032267998</MDE_IMEI>
  </USER_IDENTIFICATION>
  <STATUS>OK</STATUS>
</RESULT>

```

Figure7.XML with successful response (Result document).

Processing of the request document can be divided in two parts. The authentication part of document is processed first. If authentication is successful, the processing of the TABLE element may start. The sub element LOCATIONS is processed in case of position synchronization. The content of the LOCATIONS element is extracted to the database table named as LOCATIONS. As you can see, structure of the ROW element inside LOCATIONS element is very similar to structure of the LOCATION table (Fig 8). So it is very easy to map the row's attributes values to the database table columns in this approach. In addition with native support for XML in the Oracle database is the extraction of the elements and their values very simple.

Fig. 9 shows the flow of processing the XML document in database. In figure you can see which native Oracle XML are used to extract values from XML document. The extraction process itself does not provide any filtering of input rows. This means that each ROW element is stored in database regardless of the contained value. This is useful for troubleshooting because it is possible to completely reconstruct activity of GPS provider on mobile device. Filtering is performed at the time of selecting the data for presentation to users (see the following section).

The entire synchronization process is inspired by SAMD algorithm [9][10].

V. PRESENTATION OF POSITIONAL DATA

Once the data is stored in a central database, it is possible to display the data in well arranged graphical form. The presentation of positional data is realized through the web interface. This chapter briefly describes the utilization of collected GPS coordinates and their filtration and subsequent presentation.

Users access the information about location through the module map. This map module is completely built using Google's maps API.

In the map module, user has access to the following information:

1. The movement of the selected employees in the specific day (Fig. 10).
2. The last position of selected employees.

```

CREATE TABLE TOPS_ADM.LOCATIONS
(
  LOC_ID NUMBER(12, 0) NOT NULL PRIMARY KEY
, MDE_ID NUMBER(5, 0) NOT NULL
, USR_ID NUMBER(6, 0)
, LOC_LATITUDE NUMBER(9, 6)
, LOC_LONGITUDE NUMBER(9, 6)
, LOC_ACCURACY NUMBER(5, 0)
, LOC_PROVIDER CHAR(1 BYTE)
, LOC_SAMPLE_TIMESTAMP TIMESTAMP(0) WITH LOCAL TIME ZONE
, LOC_STATUS_MSG NUMBER(2, 0) NOT NULL
, CREATED TIMESTAMP(2) DEFAULT systimestamp(2) NOT NULL
, WORK_STATUS NUMBER(2, 0) DEFAULT 1 NOT NULL
, PROVIDER_STATUS VARCHAR2(255 CHAR)
, LOC_STATUS NUMBER(2, 0) DEFAULT 1 NOT NULL
, EVENT_MESSAGE VARCHAR2(400 CHAR))

```

Figure8.LOCATIONS table CREATE statement.

3. The addresses of the clients.
4. Pre-defined points of interest.

You can see some examples of mapping module on the following pictures.

Fig. 11 shows the last position of the selected employees. After the cursor is placed on the corresponding mark on the map, module shows the detail information about position (Fig. 12). In detail, there are listed GPS coordinates, coordinates time stamp or potential degree of inaccuracy.

A. Selection of GPS coordinates for visualization

As was mentioned before, all GPS provider data from mobile device is stored to the database. As was apparent from the previous examples, not all data from GPS provider includes the GPS coordinates (typical example is GPS provider status information). This non-positional information is not displayed to the end users and before visualization is filtered out.

But this is only the first stage of filtering. The second stage of filtering is much more sophisticated. This stage is based on analyze of the SNR³ information which is stored in the “STATUS_PROVIDED” attribute. In our system, the STATUS_PROVIDER attribute contains following data:

1. Number of fixed GPS satellites.
2. SNR value for each fixed GPS satellite.

The data is then verified by these rules:

1. Number of fixed satellites must be greater than three.
2. Number of satellites with SNR greater than 200 must be greater than three.
3. The average SNR value of all the satellites must be greater than 200.

The GPS coordinate with these parameters shows high degree of accuracy. With fewer number of satellites or lower average SNR value sometimes happens that displayed location did not correspond to the real position (inaccuracy in tens of meters).

Our method of coordination filtering is quite simple because we do not need extreme accuracy. In the situation when the extreme accuracy is required you must consider many problems associate with GPS tracking which are described in [11] [12].

The integration of Google Maps into the system was quite easy. But there is one complication about Google Maps. During the development phase of our system Google change license terms [13]. This change of the terms is focused on free usage of maps. Nowadays you can display map in a limited number of displays for free. It is no problem for us by now. But in the future we expecting a higher number of user and then it may be a problem. This may result in a change of map provider in the end.

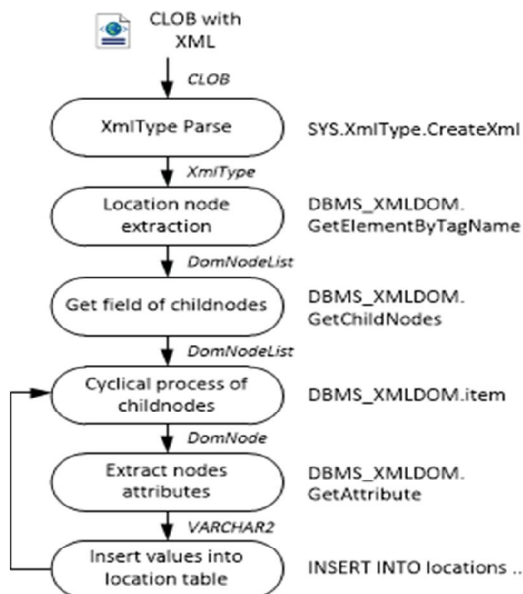


Figure9.XML document processing with built-in PL/SQL functions

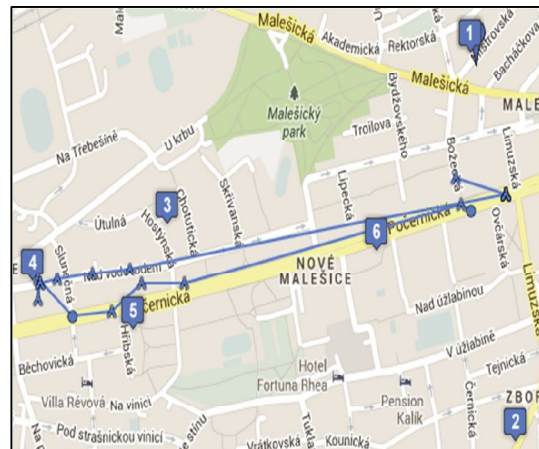


Figure10. Visualization of the track passed by worker

is based on analyze of the SNR information which is stored in the “STATUS_PROVIDED” attribute. In our system, the STATUS_PROVIDER attribute contains following data:

1. Number of fixed GPS satellites.
2. SNR value for each fixed GPS satellite.

The data is then verified by these rules:

³Signal-to-noise ratio – measure used in science and engineering that compares the level of a desired signal to the level of background noise.

1. Number of fixed satellites must be greater than three.
2. Number of satellites with SNR greater than 200 must be greater than three.
3. The average SNR value of all the satellites must be greater than 200.

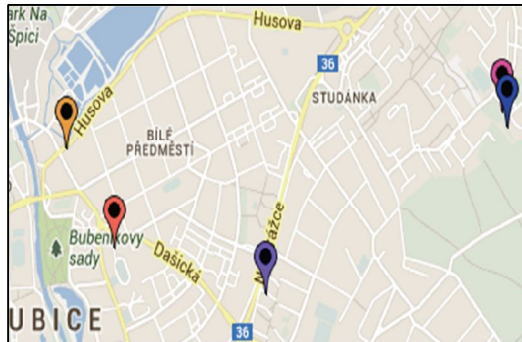


Figure 11. Last positions of the selected workers.
Each worker is distinguished by specific color.

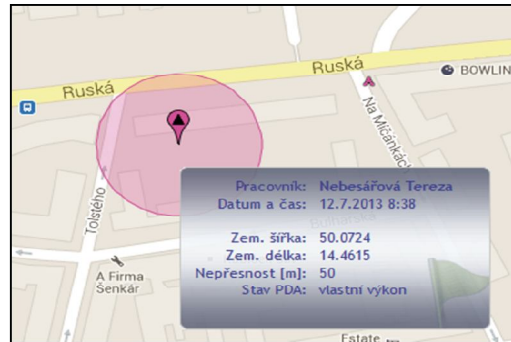


Figure 12. Detail of last position.

The GPS coordinate with these parameters shows high degree of accuracy. With fewer number of satellites or lower average SNR value sometimes happens that displayed location did not correspond to the real position (inaccuracy in tens of meters).

Our method of coordination filtering is quite simple because we do not need extreme accuracy. In the situation when the extreme accuracy is required you must consider many problems associate with GPS tracking which are described in [11] [12].

The integration of Google Maps into the system was quite easy. But there is one complication about Google Maps. During the development phase of our system Google change license terms [13]. This change of the terms is focused on free usage of maps. Nowadays you can display map in a limited number of displays for free. It is no problem for us by now. But in the future we expecting a higher number of user and then it may be a problem. This may result in a change of map provider in the end.

VI. RESULTS

The position tracking module is already more than three years in operation. More than 35 millions of rows with positional data were synchronized during that time. On average, there are more than 100 mobile devices in operation daily. Each mobile device sends between 1000 and 2000 records per day. It is interesting that 37% of all synchronized rows are the information about status of GPS provider. The remaining 63% of synchronized rows contains pure GPS coordinates. The whole LOCATION table takes more than 8 GB of disk space (include index size). This makes the LOCATION table biggest table in system so far. One of the future challenges is to increase the percentage of pure GPS coordinates in synchronization.

In the figures 13, 14 and 15 you can see some of the metrics of our system. Figures 13 show how the number of positional records growth over the three years. Figure 14 is similar but instead of number of records shows the overall size of positional data (in this case there is the net size of data without indexes). In both cases, you can see that trend is stabilized by now. Figure 15 show the average number of active mobile devices per month.

As you can imagine such a big table is performance problem. When the LOCATION table was small there was no problem. As the table grew then performance began to fall disproportionately (more than 20 second per one query). It was necessary to realize the optimization process.

We completely rewrite all queries from map module during the optimization. The whole communication architecture between web server and database system was changed. Before optimization we use the communication based on the JSON⁴ format. But we found that creation of JSON document on the database side is quite slow. So now the database sends the open database cursor and process the cursor on the web server. This way is much faster (orders of magnitude).

Furthermore, we focused on indexes. Indexes speed up the queries but with their usage is associated with some problems. First, indexes consume extra disk space and each index adds extra performance overhead for

⁴ JSON – JavaScriptObjectNotation

operation like insert, delete or update. So we were very carefully creates indexes on LOCATION table and we ends with 6 indexes on single columns.

This optimization helps speed up queries on acceptable level (1-4 seconds). In future we plan to test the table partitioning but now we are limited by used version of database system which does not support table partitioning.

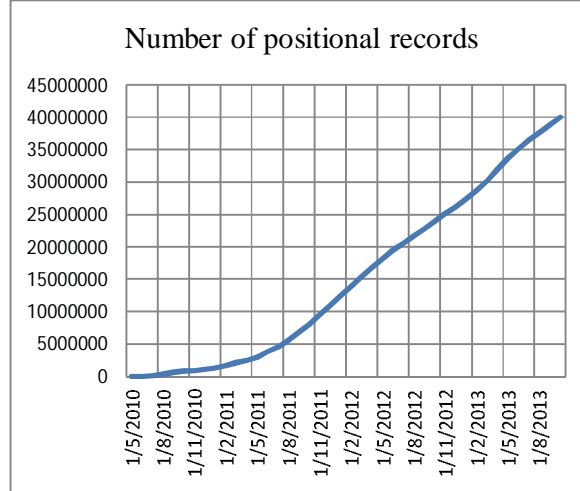


Figure13. The curve showing the number of positional records captured in last three years.

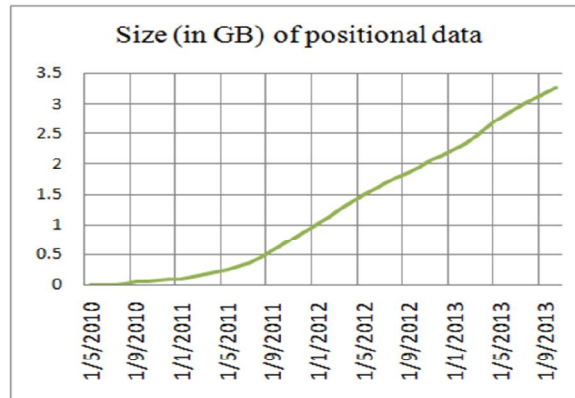


Figure14. The total amount of data collected in last three years.

VII. DISCUSSION

We have develop a tested the GPS synchronization solutions for system in production usage. We built our solution on Oracle Database 11g and Android mobile operating system. This chapter briefly summarizes some of the development challenges and chooses.

In context of similar solutions, this solution can be considered more or less unique in the way of implementation and used technologies. This is caused by two factors. Firstly, the whole development is strictly driven by specific requirements. Secondly, there was no space for selecting technologies because the solution was implemented into the existing working system. This limits our ability to draw on knowledge from other related works.

However we do not think that fewer resources and existing solutions similar to our solution would be restrictive. With less of related work we try to fully focus on own solution. Thanks to this is our solution completely built up on our own code. This helps to the code maintainability and makes the code minimalistic because there are no unused features.

Big challenge was to select an appropriate mobile platform. Finally we select the Android platform. This platform was selected due to favorable costs. Devices with Android are cheaper (cheap devices, open source IDE) at general then mobile devices with iOS or Windows. And the costs were the main attribute because the typical users of this system are the non-profit organizations financed by state. Android platform is also developer friendly.

We have verified that used combination of technologies (Android, Oracle Database and Google Maps) is appropriate for the proposed solution. We have also verified the assumption that the batch synchronization is good for overall battery life. Without batch synchronization (with on-line synchronization) we were serious problem with a battery life of mobile devices which were not capable to work more than a few hours.

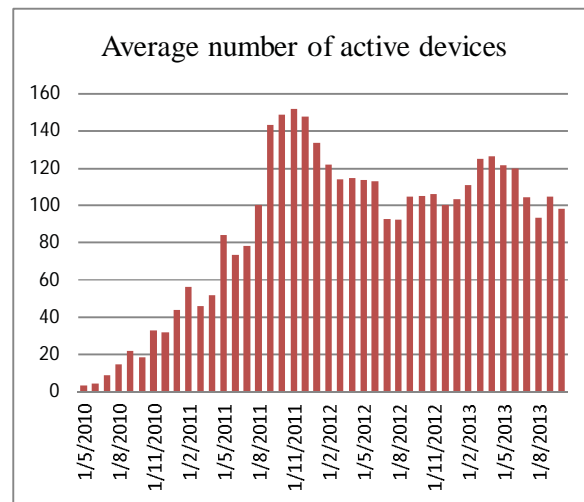


Figure15. Average number of the active devices per month.

VIII. CONCLUSION AND FUTURE WORK

The article describes one possible solution of positional data synchronization. The resulting solution is aimed to synchronization from mobile device towards the central database system. The focus is put on the exchange data format and their processing. The article can be held as methodological article when the resulting solution is coupled with a specific technology. But whole principle can be applied to various combinations of technologies.

The result of these solution is set database packages and the scheme creational script both for Oracle Database 11g (with modifications it is portable to PostgreSQL). Mobile application is also part of the resulting solution. This mobile application is developed under the Android OS.

The solution has a number of future challenges. At first, we want to improve the overall performance of the solution. There is the big challenge in processing of large amounts of data collected by GPS synchronization. Another challenge relates to power efficiency. GPS coordinates collecting is one of the most power consuming feature in our mobile subsystem. We want to shorten the time when the mobile device collecting the GPS coordinates and we want to optimize the data transfers between the mobile device and the central database system.

REFERENCES

- [1] Rigby M., Hill R., Koch S., Keeling D.: Social Care Informatics As An essential Part Of Holistic Health Care: A Call For Action. International Journal of Medical Informatics 80, 544-554 (2011). http://www.esf.org/Index.php?Eid=Tx_Nawsecuredl&U=0&file=fileadmin/Be_User/Ew_Docs/09-021_Report.Pdf&T=1291808938&Hash=7bc1e0cec1e9af27ab7d364009429ded
- [2] M. Rigby (Ed.), The Challenges of Developing Social Care Informatics as an Essential Part of Holistic Health Care: Report of an ESF Exploratory Workshop held at Keele University on 21–23 July 2010, Keele, 2010 (available at http://www.esf.org/index.php?eID=tx_nawsecuredl&u=0&file=fileadmin/be_user/ew_docs/09-021_Report.pdf&t=1291808938&hash=7bc1e0cec1e9af27ab7d364009429ded)

- [3] K. Sang-ouk, O. Se-Bong, S. Sung-Young, and L. Jin-Ho, "The framework for synchronization in embedded database environment", *Journal of Computing Science and Engineering*, vol.20, pp. 14-21, 2002.
- [4] M. Katina, C. Roger: Location and tracking of mobile devices: Ubervallance stalks the streets. *Computer Law & Security Review* 29, 216-228, 2013.
- [5] D. Barbara, "Mobile Computing and Databases - A Survey", *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, pp. 108-117, 1999.
- [6] V. Balakumar, I. Sakthidevi, "An Efficient Database Synchronization Algorithm for Mobile Devices Based on Secured Message Digest", 2012 International Conference on Computing, Electronics and Electrical Technologies, 2012.
- [7] Barabara D. "Mobile Computing and Databases – A Survey", *IEEE Transactions on Knowledge and Data Engineering*, Vol.11 No. 1, pp 108-117, 1999.
- [8] Zechmeister J., Zak D., Vana T., Lebduska J., Realization of Data Transfer between Mobile Device and Oracle Database. *Proc. of.Int. Conf on Advances in Communication and Information Technology* 2012, pp 48 – 50, 2012.
- [9] Mi-Young Choi, Eun-Ae Cho, Dae-Ha Park, Jong-YounBae, Chang-Joo Moon, Doo-Kwon Baik, A Database Synchronization Algorithm for Mobile Devices. *IEEE Transaction on Consumer Electronics*, Vol. 56, No. 2, May 2010.
- [10] Lebduska J., Zak. D., Vana T., Zechmeister J., Oracle Server-Driven Data Synchronization Using SAMD Algorithm. *Proc. of.Int. Conf on Advances in Information and Communication Technology* 2012, pp 62-66, 2012.
- [11] M. Sahmoudi, M.G. Amin: Robust tracking of weak GPS signals in multipath and jamming environments. *Signal Processing* 89, pp. 1320-1333, 2009.
- [12] R. Wang, M. Yao, Z. Cheng, H. Zou: Interference cancellation in GPS receiver using noise subspace tracking algorithm. *Signal Processing* 91, pp. 338-343, 2011.
- [13] Google Geo Developers blog: Updates to the Google Maps/Google Earth APIs Terms of Service, 2011, Available from: <http://googlegeodevelopers.blogspot.cz/2011/04/updates-to-google-maps-api-google-earth.html>
- [14] Marston S., Li Z., Bandyopadhyay S., Zhang J., Ghalsasi A., Cloud computing – The business perspective. *Decision Support Systems* 51, pp. 176-189, 2011.
- [15] Alford T., Morton G., *The Economics of Cloud Computing*. Booz Allen Hamilton, 2009.
- [16] Woody T., The future of cloud computing: Today's Weather Report [cited 2013]; Available from: <http://upstart.bizjournals.com/companies/innovation/2009/03/09/Todays-Weather-Report.html>
- [17] Zissis D., Lekkas D., Addressing cloud computing security issues. *Future Generation Computer Systems* 28, pp. 583-592, 2012.
- [18] Espada J. P., Crespo R. G., Martínez O. S., G-Bustelo B. C. P., Lovelle J. M. C., Extensible architecture for context-aware mobile web applications. *Expert Systems with Applications* 39, pp. 8686-9694, 2012.
- [19] Shen H., Reilly M., Personalized multi-user view and content synchronization and retrieval in real time mobile social software applications. *Journal of Computer and System Sciences* 78, pp. 1185-1203, 2012.
- [20] Kao Y.-W., Lin Ch., Yang K.-A., Yian S.-M., A Web-based, Offline-able, and Personalized Runtime Environment for executing applications on mobile devices. *Computer Standards & Interfaces* 34, pp. 212-224, 2012.
- [21] Hung S.-H., Shih C.-S., Shieh J.-P., Lee Ch.-P., Huang Y.-H., Executing mobile applications on the cloud: Framework and issues. *Computers and Mathematics with Applications* 63, pp. 573-587, 2012.